

Unit 1: Using Objects and Methods

User Input with Scanner

Adapted from:

1) Building Java Programs: A Back to Basics Approach
by Stuart Reges and Marty Stepp

This work is licensed under the
Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

<https://longbaonguyen.github.io>

Input and `System.in`

- **interactive program:** Reads input from the console.
 - While the program runs, it asks the user to type input.
 - The input typed by the user is stored in variables in the code.
 - Can be tricky; users are unpredictable and misbehave.
 - But interactive programs have more interesting behavior.
- **Scanner:** An object that can read input from many sources.
 - Communicates with `System.in` (the opposite of `System.out`)
 - Can also read from files, web sites, databases, ...

What Is Actually Assessed

- Be clear about what the exam does and does not ask of you here.
- Topic 1.4 says "develop code to read input," and names Scanner as ONE way to read text from the keyboard. But it carries an exclusion:
- "Any specific form of input from the user is outside the scope of the AP Computer Science A course and exam."
- Topic 4.6 is blunter still: "Accepting input from the keyboard is outside the scope."
- So keyboard Scanner code will not be tested. Scanner still matters, but for reading FILES, which is topic 4.6. The exam reference sheet lists only one constructor: `Scanner(File f)`.
- Learn this deck as the on-ramp to file reading, not as exam material in itself.

Scanner syntax

- The Scanner class is found in the `java.util` package.

```
import java.util.*;    // so you can use Scanner
```

- Constructing a Scanner object to read console input:

```
Scanner name = new Scanner(System.in);
```

- Example:

```
Scanner console = new Scanner(System.in);
```

Scanner methods

Method	Description
<code>nextInt()</code>	reads an <code>int</code> from the user and returns it
<code>nextDouble()</code>	reads a <code>double</code> from the user
<code>next()</code>	reads a one-word <code>String</code> from the user
<code>nextLine()</code>	reads a one- <i>line</i> <code>String</code> from the user

- Each method waits until the user presses Enter.
- The value typed by the user is returned.

```
prompt System.out.print("How old are you? "); //  
        int age = console.nextInt();  
        System.out.println("You typed " + age);
```

- **prompt:** A message telling the user what input to type.

Scanner example

```
import java.util.*;    // so that I can use Scanner
```

```
public class UserInputExample {  
    public static void main(String[] args) {  
        Scanner console = new Scanner(System.in);
```

```
        → System.out.print("How old are you? ");
```

age

```
        → int age = console.nextInt();
```



years

```
        → int years = 65 - age;
```

```
        System.out.println(years + " years to retirement!");
```

```
    }
```

```
}
```

- Console (user input underlined):

How old are you 29

36 years until retirement!



Input tokens

- **token**: A unit of user input, as read by the Scanner.
 - Tokens are separated by *whitespace* (spaces, tabs, new lines).
 - How many tokens appear on the following line of input?
23 John Smith 42.0 "Hello world" \$2.50 " 19"

- When a token is not the type you ask for, it crashes.

```
System.out.print("What is your age? ");  
int age = console.nextInt();
```

Output:

```
What is your age? Timmy  
java.util.InputMismatchException  
    at java.util.Scanner.next(Unknown Source)  
    at java.util.Scanner.nextInt(Unknown Source)  
    . . .
```

Scanner example 2

```
import java.util.*;    // so that I can use Scanner

public class ScannerMultiply {
    public static void main(String[] args) {
        Scanner console = new Scanner(System.in);

        System.out.print("Please type two numbers: ");
        int num1 = console.nextInt();
        int num2 = console.nextInt();

        int product = num1 * num2;
        System.out.println("The product is " + product);
    }
}
```

- Valid Outputs (user input underlined):

Please type two numbers: 8 6

The product is 48

// 2 tokens separated by space

Please type two numbers: 8

6

The product is 48

// 2 tokens separated by new
// line

Reading from a Text File

- A Scanner does not have to read the keyboard. Hand it a File instead and it reads from that file, using the same methods you already know.

```
import java.io.File;
import java.util.Scanner;
Scanner input = new Scanner(new File("data.txt"));
while (input.hasNext()) {
    String line = input.nextLine();
    System.out.println(line);
}
input.close();
```

- hasNext() returns false once nothing is left to read, so it is what ends the loop. close() releases the file when you are done.
- A method that opens a file must declare throws IOException in its header. Reading files is on the exam; writing files is not.

Strings as user input

- Scanner's next method reads a word of input as a String.

```
Scanner console = new Scanner(System.in);  
System.out.print("What is your name? ");  
String name = console.next();  
System.out.println("Your name is " + name);
```

Output:

```
What is your name? Chelsey  
Your name is Chelsey.
```

- The nextLine method reads a line of input as a String.

```
System.out.print("What is your address? ");  
String address = console.nextLine();  
System.out.println("Your address is " + address);
```

Output:

```
What is your address? 123 Fake st.  
Your address is 123 Fake st.
```